Appendix A: Tiling selection code

In order to track the highest scoring tiling along with its score, let us associate a set of tiles and a score in an ordered pair $s = (a, \mathcal{T})$. Here, a is the score and $\mathcal{T}$ is a (possibly partial) tiling, meaning that it is a set of non-overlapping tiles. When the tiling is partial, the tiles do not tile the entire space. The notation s_{score} and s_{tiling} will be used to refer to the elements of the ordered pair. The notation $T(x, R_L, R_H, C_L, C_H)$ will be used to represent a tile with outbreak state x (0 or 1) and boundaries R_L, R_H, C_L, C_H. $\emptyset$ denotes the empty set.

Tiling selection algorithm:

```
1    For a grid of size R × C
2    Track best hypotheses: bestR, bestC
3    Keep caches rCache, cCache
4    Init:
5        rCache[0] ← (1, ∅)
6        cCache[0] ← (1, ∅)
7    for R_H from 1 to R
8        bestR ← (0, ∅)
9        for R_L from 1 to R_H
10           for C_H from 1 to C
11               bestC ← (0, ∅)
12               for C_L from 1 to C_H
13                   x ← argmax_{x∈{0,1}} score(x, R_L, R_H, C_L, C_H)
14                   s ← (score(x, R_L, R_H, C_L, C_H), {T(x, R_L, R_H, C_L, C_H)})
15                   if bestC_score < cCache[C_L − 1]_score · s_score then
16                       bestC ← (cCache[C_L − 1]_score · s_score, cCache[C_L − 1]_tiling ∪ s_tiling)
17                   end if
18               end for
19               cCache[C_H] ← bestC
20           end for
21           if bestR_score < rCache[R_L − 1]_score · bestC_score then
22               bestR ← (rCache[R_L − 1]_score · bestC_score, rCache[R_L − 1]_tiling ∪ bestC_tiling)
23           end if
24       end for
25       rCache[R_H] ← bestR
26   end for
27   return bestR
```